Ground Penetrating Radar Data Acquisition System

Robert Beck, Jay Cosentino, David W. Collier, and Jim Osborn

CMU-RI-TR-91-11

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

June 1991

Table of Contents

List of Figures

Abstract

Carnegie Mellon University is automating the use of Ground Penetrating Radar (GPR) for cleanup of hazardous waste sites. The Site Investigation Robot (SIR) project at the Field Robotics Center is applying robotics and image processing technologies to the investigatory phase of these waste site cleanups. The current focus is on the development of an automated subsurface mapping system to locate buried objects and geological structures so that sources and migratory pathways of contaminants can be identified and cataloged. The subsurface maps are produced using the non-invasive sensing abilities of Ground Penetrating Radar. GPR operates on principles similar to conventional radar, but the data acquired is more difficult to process due to the heterogeneous nature of the subsurface medium. GPR deployment, data acquisition, and interpretation are traditionally human-driven processes which expose operators to potentially dangerous environments. Automating the GPR data collection process eliminates this undesirable situation. Accurate three dimensional subsurface maps of GPR data have not yet been generated in the field. However, the SIR project uses robots to position GPR transducers to exploit the accurate, repeatable positioning available from automated equipment. By combining the use of a position-cognizant, all-terrain mobile robot and a linear scanning mechanism, it is possible to acquire GPR records in a two-dimensional grid on the ground surface.

1.0 System Overview

The Terregator (Figure 1) is a six-wheeled, skid-steered mobile robot which is used to position a GPR transducer. The transducer is suspended from a wheeled cart, or "buggy," which rolls along a one-axis gantry attached to the vehicle. During data acquisition the buggy is moved in a direction orthogonal to the Terregator's path. Data is collected at incremental locations in a predetermined grid on the hazardous waste site. The GPR transducer contains a matched antenna pair for transmission and reception of radar signals. High-frequency radar pulses transmitted by the send antenna are reflected back to the receive antenna by interfaces between subsurface materials with different electromagnetic characteristics. Received reflections are only stored for predetermined site locations.

2.0 Detailed System Description

The system is divided into four areas: GPR data acquisition hardware, robotics, communications, and computing. The onboard computing controls the robot's motion and the collection of GPR data based on high-level commands which are specified through a graphical user interface on a remote workstation. These commands are sent via a radio modem from the workstation to the robot, where they are translated into low-level commands and executed. The GPR unit transmits radar waves and receives subsurface reflections which are digitized by the computing equipment and stored on a hard disk.

2.1 GPR Data Acquisition Unit

The GPR unit is a Geophysical Survey Systems Incorporated (SIR-3) containing a model 3102 transducer and a model PR-8300 profiling recorder. The transducer contains two antennas tuned to a center frequency of approximately 500 Mhz, the circuitry to translate power to high frequency radar waves for transmission, and the circuitry to transform received high frequency radar waves to audio frequency. Its dimensions are 15 by 31 by 36 centimeters and weighs 4 kilograms. Configuration of the antenna, necessary before beginning data collection at a site, is accomplished by adjusting switches, dials, and potentiometers on the PR-8300. Its dimensions are 45 by 45 by 22 centimeters and weighs 2.7 kilograms. The unit is powered by 15 Amps at 12.5 volts DC.

Refer to Figures 2 and 3 for the following explanation. Analog data from the GPR transducer is conditioned and digitized before it can be used by the computer or stored on the hard disk. The transmitting antenna outputs pulses of a 500 MHz waveform. This pulsed waveform is recovered by the receiving antenna after reflection from subsurface objects. Because of the high frequency of the output waveform, it is not practical to sample an entire reflected pulse. Instead, a series of pulses is output at each position, or "scan point", at which data is being acquired. The reflections from this series of output pulses are assumed to be identical, since they all represent the same area of ground. The first reflected pulse is sampled immediately. The second pulse is sampled 50 ps after it is received. The next pulse is sampled 100 ps after it is received, and so on. The samples therefore occur at the rate at which the pulses are output, 50 kHz, but they represent data points which are actually 50ps apart (an "effective" rate of 20 GHz). The PR-8300 outputs the sampled data as groups of analog output pulses (see Figure 3, signal 3). The PR-8300 also outputs a start of scan point signal which indicates the first received pulse in a series for a given position on the surface.

In the next step, the GPR interface board conditions the analog pulses so that they can be read by the analog to digital (A/D) converter on the DAADIO board. The pulses are output by the PR-8300 in groups, where each group represents one sample of the receiving antenna. Therefore, all of the pulses in a group should be the same magnitude. Because of the nature of the sampling that creates these pulses, some of the pulses in each group are noisy. Since each group of pulses only represents one data point, the noisy pulses are eliminated and the best pulse in each group is passed on to the DAADIO board. The middle pulse of each group is the least noisy, so this is the one which is selected by the GPR interface board. The GPR interface board also creates a trigger signal for the DAADIO board. This is a digital signal which is high whenever an analog pulse is available for A/D conversion. The GPR interface board also converts the start of scan point signal to a level which is acceptable for input to the DAADIO board.

The maximum time resolution of the data is 50 picoseconds. Scan points are taken at regular intervals - the default is 16 points per second, but other settings are available on the PR-8300. As the buggy moves the antenna along the gantry, the CPU reads GPR data points at incremental positions in a prespecified grid. All of the scan points acquired in one move of the buggy are called a scan. When studying a displayed two dimensional scan of GPR data, it is important to know the location (x, y, time) of any reflections. A data pulse - on the digital data line between the DAADIO board and the CPU - represents radar reflections from a position (x,y,time) in the subsurface. This point in the subsurface is defined by the time the radar waves take to travel from the send antenna to the subsurface position and return to the receive antenna. When the polling algorithm stores every data item, the maximum time resolution is achieved. Storing every other data point would decrease the resolution by half. Conversion of time to depth is based on the velocity of the radar waves in the medium.

Scans are displayed on a color workstation screen in a rectangular area. The entire range of collected data values is mapped to a user specified range of colors (e.g. - blue to green). The data represented by the colors in one column on the display corresponds to one scan point. Depth ranges from zero (column top) to a maximum value (column bottom). Surface distance is measured along the display's horizontal axis. When the computer mouse is clicked on a point in the display, the position where that data item was acquired is printed in the top left corner of the image.

2.2 Robotics

The Terregator's dimensions are 1.6 by 1.1 meters, and the gantry's dimensions are 2.8 by 0.5 meters. Its motion is achieved by a combination of straight line segments and zero-radius turns, also known as point turns. During straight line moves, the motion control routines use simple dead reckoning to calculate changes in the vehicle's global position based on incremental wheel encoder counts and the starting position of the vehicle. Since the Terregator skids while turning, feedback from the wheel encoders is combined with angular measurements from the gyroscope to correct for these errors.

The buggy, used to position the GPR transducer, is pulled along the gantry by a chain which is driven by a geared wheel attached to the shaft of a motor. Changes in the buggy's position are read from an incremental wheel encoder at the motor shaft. A calibration routine employing feedback from limit switches at the ends of the gantry is used to establish the buggy's absolute initial position.

Before acquiring data at a site, the robot is driven to a place of known position called the calibration point. The position of the site and its orientation relative to the calibration point are known. The Terregator's planning routine divides the surface of the site into a number of rectangular patches (Figure 4), and a path is calculated around the site. The path is described by a series of motion and data acquisition commands which are sent to the Terregator (via a modem) and used to command the vehicle to collect data at each surface patch. Within a surface patch, the robot moves in steps of equal size, from one side of the patch to the other. At each stopping point the GPR antenna "scans" the ground in a direction orthogonal to the robot's path. Each surface location where a radar waveform is digitized is called a "scan point".

2.3 Communications

Communication between the Terregator and a remote workstation is accomplished via a 1200 baud radio modem (Vectran VT-30). Commands are sent from the workstation to the Terregator based on input from a user interface. These commands are of four types:

- Motion commands: Low level commands used to change the Terregator's position.
- Scan commands: A set of high level commands used to set up and execute all the vehicle moves, gantry moves, GPR data collection, and data storage routines needed to collect data for a surface patch.
- System commands: Low level commands used to initialize computing equipment and servo amps.
- Query commands: Low level command used to get information such as current position from the Terregator.

An "acknowledge" signal is sent from the Terregator to the workstation to verify receipt of each command. After each command is executed, a "done" message is sent from the Terregator to the workstation. A terminal on the Terregator is used to display error messages and other diagnostic information.

2.4 Computing

The Terregator's computing consists of all the hardware and software housed in the Terregator VME cage (Figure 2). The software commands a number of devices which control the robot's motion, communicate with the remote workstation, and collect data from the GPR data acquisition unit. Motion control and data acquisition software execute on the Heurikon (HKV2E) central processing unit (CPU) board; the CPU chip on this board is a Motorola 68020. An ethernet connection between the CPU and a workstation allows storage of the GPR data in a UNIX file system on hard disk. Two motion control boards (SC900V) are used to position the Terregator's wheels and the gantry buggy. The GPR unit is an analog device, requiring the use of the DAADIO board for A/D conversion of data. The DAADIO board is also used for digital switching (including the activation of servo amplifiers) and enabling of limit switch signals. Three devices are connected to the CPU through its serial ports: an on-board terminal, a gyroscope, and a radio modem.

The implementation language is "C", running under the VxWorks real time operating system. The GPR data resides in "C" data structures in memory before being copied into a custom file format on hard disk (Figure 3, 4, and 5). The explanation of the data structures is divided into two parts: a sample scenario describing the use of the structures in GPR data collection and a detailed explanation of each structure's fields. The first step in the sample scenario is to drive the Terregator to the robot calibration point. Through interaction with the remote user interface, the user specifies the area of ground over which GPR data will be collected (site). This is done by using a computer mouse to "draw" the borders of the desired site on the workstation screen, in a manner similar to computer aided drafting packages. Path planning software uses this specification of the site boundaries and a priori knowledge of obstacles in the site to plan a path for the robot. Traditional path planning software was developed for applications that require the shortest path between two points. Hazardous waste site remediation requires that data be acquired over as much of the site as possible, for this reason the path planning software for the SIR project maximizes the area in data acquisition. The output of the path planning software is a list of points in the precalculated site grid, each point defines the robot's starting or ending point in a surface path. The next step is to use these points to generate a series of supervisory level scan commands which acquire data over an entire surface patch. These commands are sent to the Terregator via the modem and translated into low level commands to the motion control boards and the A/D board. The first supervisory level move command instructs the robot to move to the start of the site. At this point, information about the site world position and bounding volume is stored in the site memory structure. Collection of a volume of data is divided into a series of standard steps; therefore, the methodology for storing volume data is only described for one volume. The robot moves to the start of the surface patch and writes a volume header structure to disk. Collection of a scan of data is accomplished as the transducer is moved from one side of the gantry to the other. At evenly spaced locations in the gantry movement, the CPU samples the GPR data arriving from the DAADIO board and stores it into a buffer. This buffer is stored in a scan point header. After the gantry movement is finished, all the scan points are written to one scan file on disk. The collection and storage of every scan is accomplished in the same way. When all the scans in a volume have been collected, the robot moves to the next surface patch in its predetermined path.

Refer to Figures 4 and 5 in the description of the structure fields which follows. The world position structure contains two items: the cartesian coordinates of a point and its orientation with respective to some reference point (e.g. - the reference point for the world position of a site would be the robot calibration point. A bounding volume is a rectangular solid that contains all points in a volume of GPR data; its fields are the cartesian coordinates of two points. Cataloging of GPR data is done in a hierarchical manner using four GPR structures: point, scan, volume, and site. A header is associated with each level in the hierarchy. At each position in the precalculated grid where GPR data is collected, the data is stored in a scan point structure. A scan point consists of a data buffer and a header. The size of the data buffer is dependent on the maximum range of the antenna and the electrical characteristics of the subsurface; for example, a time range of 10 nanoseconds in sand requires a buffer of length 320. The scan point header fields are a pointer to GPR data, the length of the data buffer, and the distance from the last scan point collected in the same scan. A scan consists of all scan points collected by one motion of the GPR transducer along the gantry. The fields in a scan structure are an array of pointers to scan point structures, the number of scan points, the bounding volume points, and the date collected. All the scans of data collected in one surface patch are stored in a volume structure. Its fields are an array of pointers to all scan structures, the number of scans, the world position, the bounding volume, the date collected, the time step of the data, the base permittivity of the subsurface, and title. All the volumes of GPR data collected in a site are catalogued using a site header. The fields in a site header are an array of pointers to volume headers, the number of volumes, the world position, the bounding volume, an array of pointers to names of directories where the volumes are stored, an array of pointers to volume titles, a pointer to a site title, and a pointer to user notes.

3.0 System Status

The current status of the implementation is described next. The Terregator collects GPR data from a parking lot near our laboratory and stores the data on the SUN workstation's onboard SCSI disk drive. Next, the SUN is connected to the lab ethernet using the Network File System (NFS) protocol. Processing and display of the data can now be done on a SUN Sparcstation. The path planner and graphical user interface have not been fully tested.

4.0 Conclusion

A complete system for acquiring GPR data, tagging it with its position in cartesian space, and storing it on disk has been described. It consists of a robot, the Terregator, which traverses a hazardous waste site, positions itself at points in a pre-calculated grid, and a gathers subsurface images using a GPR system. The computing equipment on board the Terregator controls the motion of the robot and the acquisition of the GPR data. Communications with a remote workstation are performed using a wireless modem.

Figure 1 - Terregator

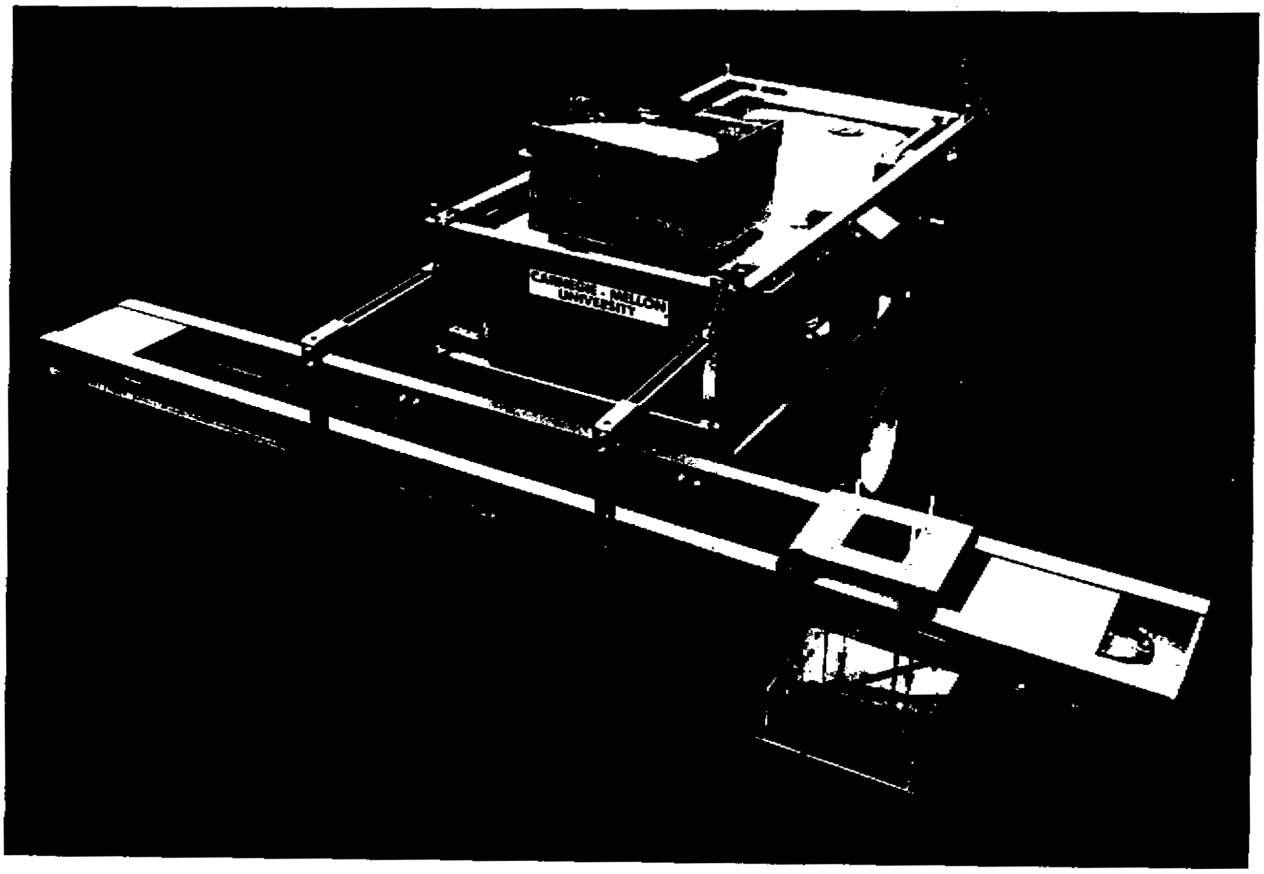


Figure 2 - Data Acquisition System

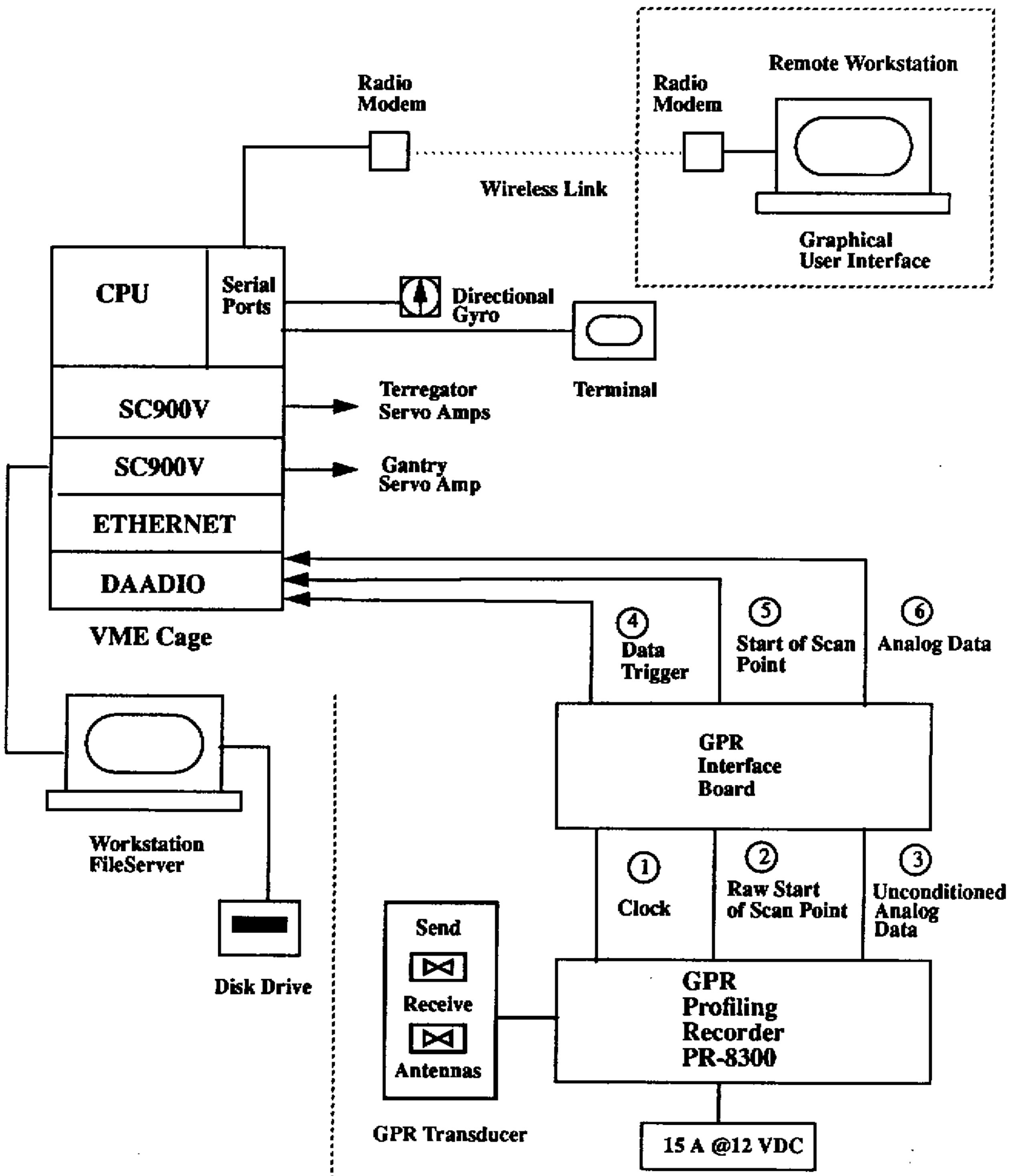
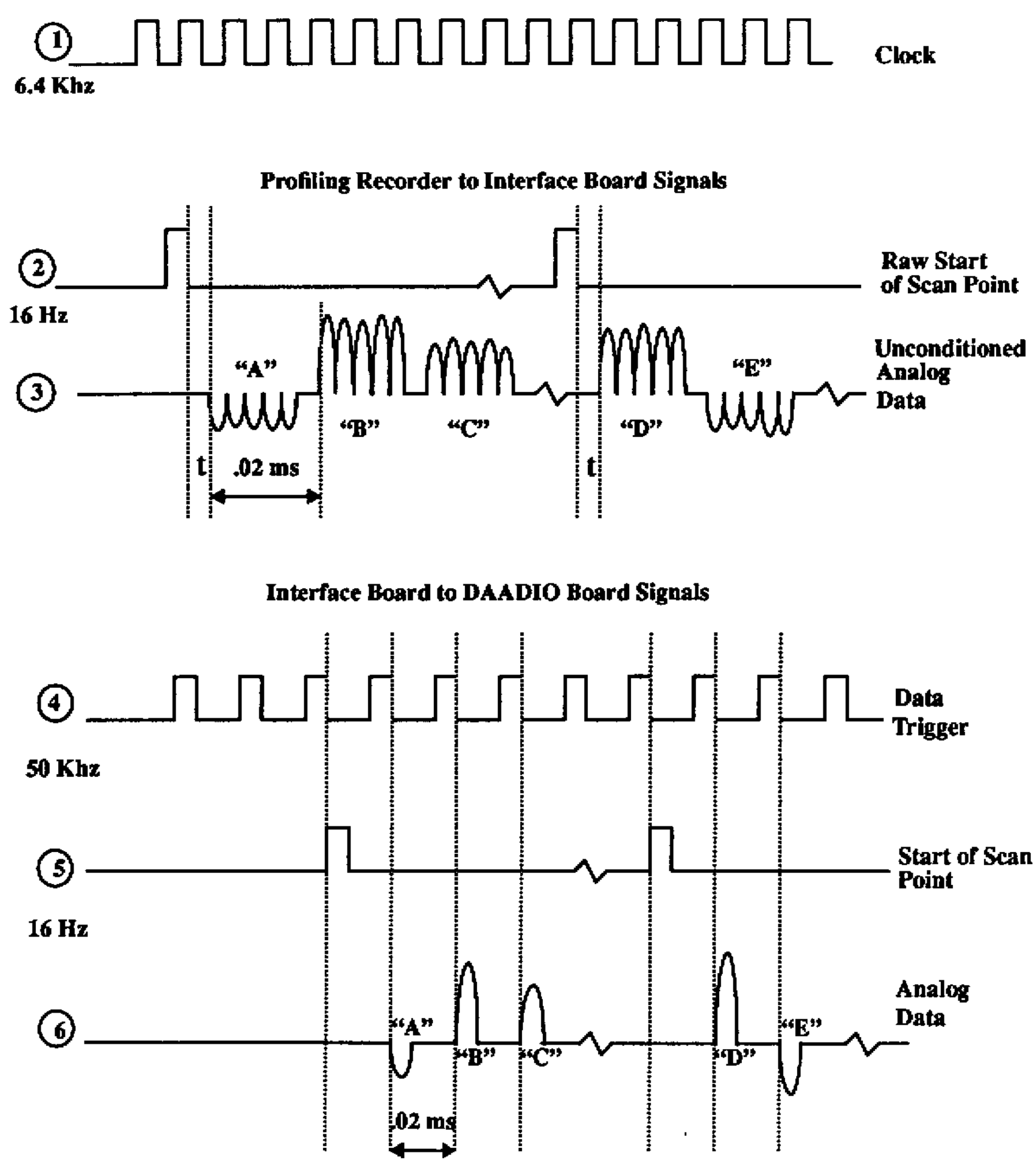


Figure 3 - Data Acquisition System Timing Diagram



Refer to Figure 2
for location of pulses on
Data Acquisition Diagram

Figure 4 - Site Terminology

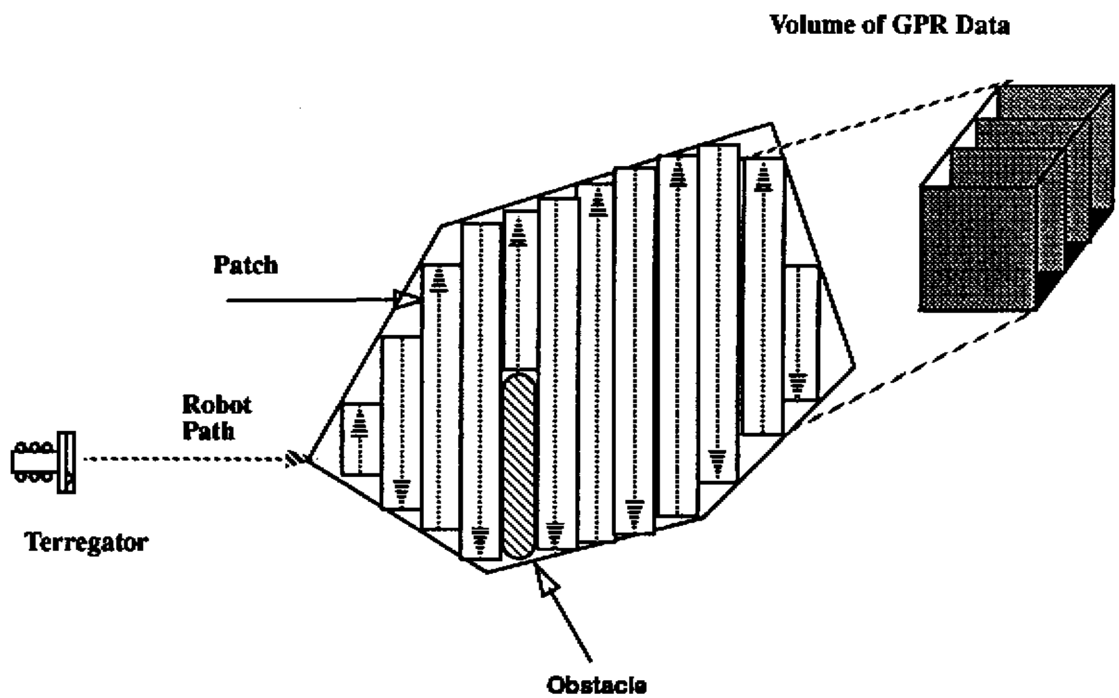


Figure 5 - Organization of GPR Structures

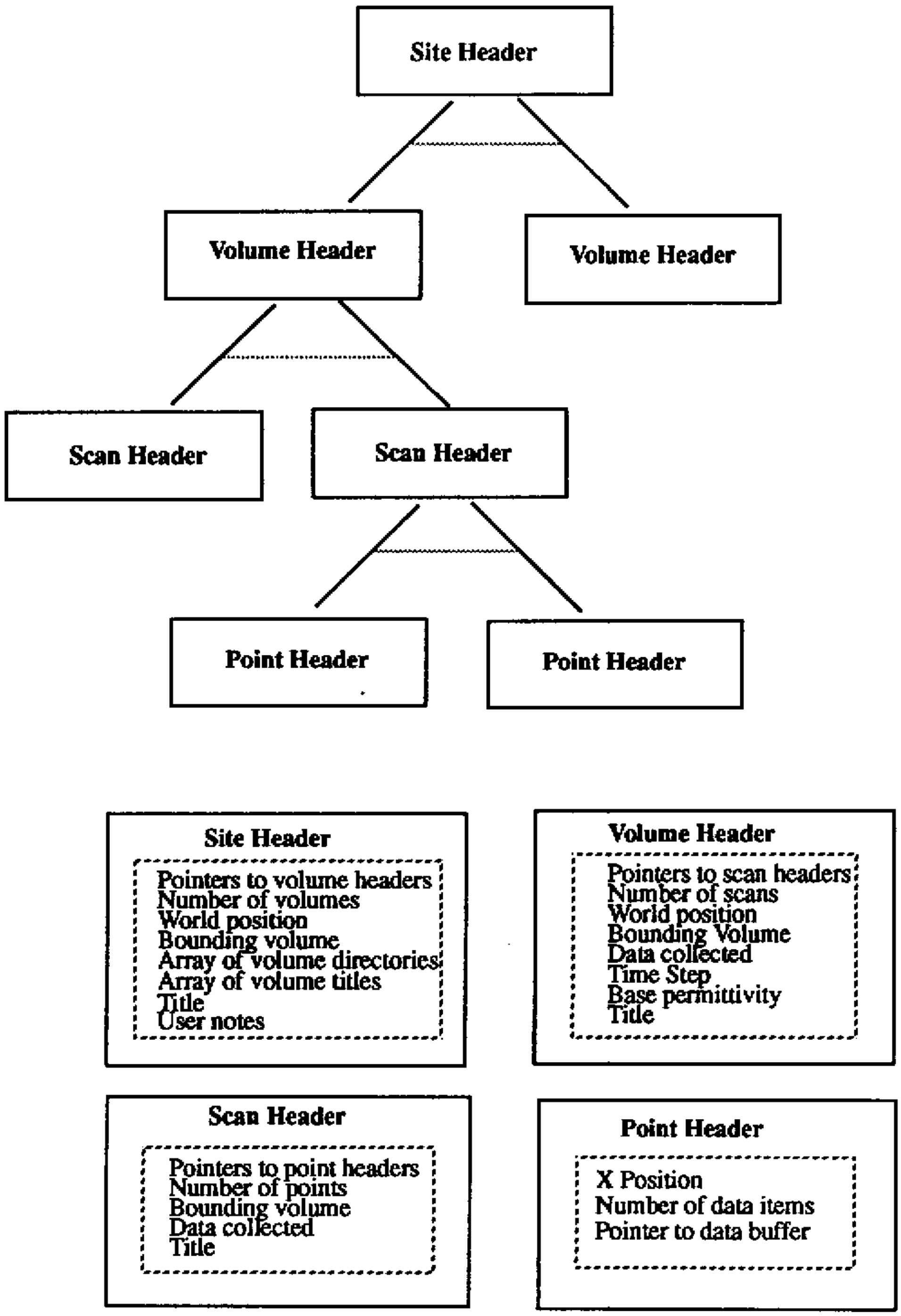


Figure 6 - GPR Structures

Code	Comments
	Miscellaneous
typedef long DATETIME_TYPE	Date and Time: UNIX format
typedef float GPR_DATA_TYPE	GPR data
typedef long POSITION_TYPE	Millimeters
typedef long DISTANCE_TYPE	Millimeters
typedef long TIME_TYPE	Time Step of the radar antenna (picoseconds)
typedef float CONDUCT_TYPE	Conductivity of the medium (mho)
typedef float ANGLE_TYPE	Radians
typedef float PERM_TYPE	Permittivity
typedef struct _BOUND_VOL {	**Bounding Volume Structure**
POSITION_TYPE min_x, min_y, min_z	Minimum & maximum x,y,z values in a volume
POSITION_TYPE max_x, max_y, max_z	
} BOUND_VOL_TYPE	
	World Position Structure
typedef struct {	Explanation: world position, relative to a reference point
ANGLE_TYPE rx, ry, rz	Roll, pitch, and yaw
POSITION_TYPE x, y, z	x, y, z position in mm
} WORLD_POS_TYPE;	
typedef struct {	**GPR Point Structure**
GPR_DATA_TYPE *data	Pointer to buffer of GPR data
int data_size	Size of the data buffer
DISTANCE_TYPE dx	Distance in the x direction from the start of the scan
} GPR_POINT_TYPE	
	GPR Scan Structure
typedef struct {	
GPR_POINT_TYPE **pointsIndex;	Array of pointers to gpr point structures
int slice_size;	Number of points in this scan
BOUND_VOL_TYPE bvol;	Bounding volume
DATETIME_TYPE date_coll;	Date collected
char *title;	Title of the slice
} GPR_SCAN_TYPE;	

Code	Comments
typedef struct {	**GPR Volume Header**
GPR_SCAN_TYPE **slicesIndex;	Array of pointers to gpr scan structures
int vol_size;	Number of points in this slice
WORLD_POS_TYPE wpos;	World position relative to the start of the site
BOUND_VOL_TYPE bvol	Bounding volume
DATETIME_TYPE date_coll;	Date collected
TIME_TYPE time_step;	Time per data entry in each data array
PERM_TYPE base_perm;	Base permittivity of the soil
char*title;	Title
} GPR_VOL_TYPE;	
typedef struct {	**Site Index Header**
GPR_VOL **volsIndex;	Array of ptrs to GPR volume headers in memory
int site_size;	Number of volumes
WORLD_POS_TYPE wpos;	World position
BOUND_VOL_TYPE bvol;	Bounding volume
char **volume_dir;	Array of volume directory names
char **volume_title;	Array of volume titles
char *title;	Site title
char *notes;	User notes
} SITE_INDEX_TYPE;	